

ON PARAMETERIZED ONE-COUNTER AUTOMATA

BENEDIKT BOLLIG, KARIN QUAAS, AND ARNAUD SANGNIER

ABSTRACT. We show that reachability in parameterized one-counter automata can be reduced to nonemptiness of two-way finite automata.

1. INTRODUCTION

In [1], Demri and Sangnier showed that model checking flat freeze LTL on one-counter automata can be reduced to generalized repeated reachability in parameterized one-counter automata. Decidability of the latter was left as an open problem. Recently, Lechner et al. showed decidability by reduction to satisfiability in Presburger arithmetic [2]. As a corollary, they obtain a 2NEXPTIME upper bound for model checking flat freeze LTL when counter updates are encoded in binary.

In this note, we demonstrate that generalized repeated reachability in parameterized one-counter automata with *unary* counter updates can be reduced, in linear time, to nonemptiness of two-way finite automata (which is PSPACE-complete). As a corollary, we obtain that model checking flat freeze LTL on one-counter automata is in EXPSpace when counter updates are encoded in unary. It is not clear, however, whether one can obtain our upper bound (for unary encoding) using the approach in [2] (for binary encoding), or vice versa.

2. PARAMETERIZED ONE-COUNTER AUTOMATA

A parameterized one-counter automaton is a classical one-counter automaton extended by parameterized equality and disequality tests of the form $\text{eq}(x)$ and $\text{neq}(x)$, respectively. Here, x is a variable that will be instantiated by a natural number.

Definition 1. A *parameterized one-counter automaton* (POCA for short) is a tuple $\mathcal{P} = (Q, X, q_0, \Delta, \mathcal{F})$ where

- Q is a finite set of *states*,
- X is a finite set of *variables*,
- $q_0 \in Q$ is the *initial state*,
- $\Delta \subseteq Q \times \text{Tests}(X) \times \{1, 0, -1\} \times Q$ is the set of *transitions* where $\text{Tests}(X) := \{\text{true}, \text{zero}\} \cup \{\text{eq}(x) \mid x \in X\} \cup \{\text{neq}(x) \mid x \in X\}$ is the set of counter tests and $\{1, 0, -1\}$ is the set of counter updates, and
- \mathcal{F} is a set of subsets of Q , representing the acceptance condition (generalized repeated reachability).

Note that we assume a unary encoding of counter updates (in terms of 1 and -1).

The semantics of \mathcal{P} is defined wrt. a variable instantiation $\gamma : X \rightarrow \mathbb{N}$ as an infinite transition system. A configuration of the transition system is a pair $(q, i) \in Q \times \mathbb{N}$ where q is the current state and i is the current counter value. We define a transition relation $\Longrightarrow_\gamma \subseteq (Q \times \mathbb{N})^2$ as follows. For two configurations (q, i) and (q', i') , we have $(q, i) \Longrightarrow_\gamma (q', i')$ if there are $\tau \in \text{Tests}(X)$ and $\sigma \in \{1, 0, -1\}$ such that $(q, \tau, \sigma, q') \in \Delta$, $i' = i + \sigma$, and one of the following holds:

- $\tau = \text{true}$,
- $\tau = \text{zero}$ and $i = 0$,
- $\tau = \text{eq}(x)$ and $i = \gamma(x)$, or
- $\tau = \text{neq}(x)$ and $i \neq \gamma(x)$.

A γ -run of \mathcal{P} is a sequence $(q_0, i_0) \Longrightarrow_\gamma (q_1, i_1) \Longrightarrow_\gamma (q_2, i_2) \Longrightarrow_\gamma \dots$ such that $i_0 = 0$. It is accepting if, for each $F \in \mathcal{F}$, there are infinitely many $j \in \mathbb{N}$ such that $q_j \in F$.

The *generalized repeated reachability problem* asks whether, given a POCA \mathcal{P} , there is an accepting γ -run of \mathcal{P} , for some γ .

3. TWO-WAY GENERALIZED BÜCHI AUTOMATA

We reduce generalized repeated reachability in a POCA \mathcal{P} , say, with set of variables X , to nonemptiness of a two-way finite automaton with generalized Büchi acceptance condition. The main idea is to encode a variable instantiation $\gamma : X \rightarrow \mathbb{N}$ as the word $w_\gamma := A_0 A_1 A_2 \dots \in (2^X)^\omega$ where, for all $i \in \mathbb{N}$, we let $A_i = \{x \in X \mid \gamma(x) = i\}$. For example, if $X = \{x_1, x_2, x_3\}$ and $\gamma = \{x_1 \mapsto 0, x_2 \mapsto 2, x_3 \mapsto 2\}$, then $w_\gamma = \{x_1\} \emptyset \{x_2, x_3\} \emptyset^\omega$. We call a word $w \in (2^X)^\omega$ a *parameter word* if $w = w_\gamma$ for some γ . In the following, we show how to transform \mathcal{P} into a two-way finite automaton that, for all γ , accepts w_γ iff \mathcal{P} has an accepting γ -run.

Given a set X , let $\overline{X} := \{\overline{x} \mid x \in X\}$ be a disjoint copy of X . Intuitively, \overline{x} can be interpreted as the negation of x .

Definition 2. A *two-way finite automaton with generalized Büchi acceptance condition* (2WFA) is a tuple $\mathcal{A} = (Q, X, q_0, \Delta, \mathcal{F})$ where

- Q is the finite set of *states*,
- X is a finite alphabet,
- $q_0 \in Q$ is the *initial state*,
- $\Delta \subseteq Q \times (\{\text{true}, \text{first}\} \cup X \cup \overline{X}) \times \{1, 0, -1\} \times Q$ is the transition relation, and
- $\mathcal{F} \subseteq 2^{2^Q}$ is the acceptance condition, defined as for POCAs.

While, in a POCA, 1 and -1 are interpreted as *increment* and *decrement the counter*, respectively, their interpretation in a 2WFA is *go to the right* and *go to the left in the input word*.

With \mathcal{A} and an arbitrary word $w = A_0 A_1 A_2 \dots \in (2^X)^\omega$, we associate a global step relation $\Longrightarrow_w \subseteq (Q \times \mathbb{N})^2$. Let $(q, i), (q', i') \in Q \times \mathbb{N}$. Then, we have $(q, i) \Longrightarrow_w (q', i')$ if there are τ and σ such that $(q, \tau, \sigma, q') \in \Delta$, $i' = i + \sigma$, and one of the following holds:

- $\tau = \text{true}$,
- $\tau = \text{first}$ and $i = 0$,
- $\tau = x \in X$ and $x \in A_i$, or
- $\tau = \overline{x} \in \overline{X}$ and $x \notin A_i$.

Similarly to POCAs, a *run* of \mathcal{A} on w is a sequence $(q_0, i_0) \Longrightarrow_w (q_1, i_1) \Longrightarrow_w (q_2, i_2) \Longrightarrow_w \dots$ where $i_0 = 0$. Acceptance is defined just as for POCAs. The language $L(\mathcal{A}) \subseteq (2^X)^\omega$ is the set of words w such that there is an accepting run of \mathcal{A} on w .

Theorem 3 (Serre [3]). *Given a collection of 2WFAs $\mathcal{A}_1, \dots, \mathcal{A}_n$ over the same alphabet X , checking whether $L(\mathcal{A}_1) \cap \dots \cap L(\mathcal{A}_n) \neq \emptyset$ is in PSPACE.*

Remark 1. The result from [3] is in fact stronger, as it is valid for *alternating* 2WFAs. Note that we use alternation in a very restricted form in terms of language intersection. Also note that we defined 2WFAs in a way that will be convenient for the translation of POCAs into 2WFAs as explained in the next section. In particular, the actual alphabet 2^X is of exponential size. However, a subset of X can be encoded as a sequence of length $|X|$ over the alphabet $\{0, 1\}$. That is, we can encode a word $\{x_1\}\emptyset\{x_2, x_3\}\emptyset^\omega$ as 100 000 011 (000) $^\omega$. This allows us to reduce our model to the one from [3] in polynomial time. Finally, [3] uses a parity acceptance condition, but it is standard to transform a generalized Büchi acceptance condition into a parity condition in polynomial time.

It is worth noting that, in [3], Serre also used two-wayness to simulate one-counter automata, but in a game-based setting (the latter is reflected by alternation).

4. FROM POCAs TO 2WFAs

The reduction of POCAs to 2WFAs is established by the following lemma.

Lemma 1. *Let $\mathcal{P} = (Q, X, q_0, \Delta, \mathcal{F})$ be a POCA. We can construct, in linear time, a 2WFA $\mathcal{A} = (Q, X, q_0, \Delta', \mathcal{F})$ such that, for all instantiations $\gamma : X \rightarrow \mathbb{N}$, the following statements are equivalent:*

- \mathcal{P} has an accepting γ -run;
- $w_\gamma \in L(\mathcal{A})$.

Proof. We only have to adapt the transition relation:

$$\begin{aligned} \Delta' = & \{(q, \text{true}, \sigma, q') \mid (q, \text{true}, \sigma, q') \in \Delta\} \\ & \cup \{(q, \text{first}, \sigma, q') \mid (q, \text{zero}, \sigma, q') \in \Delta\} \\ & \cup \{(q, x, \sigma, q') \mid (q, \text{eq}(x), \sigma, q') \in \Delta\} \\ & \cup \{(q, \overline{x}, \sigma, q') \mid (q, \text{neq}(x), \sigma, q') \in \Delta\} \end{aligned}$$

Correctness of \mathcal{A} in the sense of the lemma follows from the observation that, for all $\gamma : X \rightarrow \mathbb{N}$ and all $(q, i), (q', i') \in Q \times \mathbb{N}$, we have $(q, i) \Longrightarrow_\gamma (q', i')$ in \mathcal{P} iff $(q, i) \Longrightarrow_{w_\gamma} (q', i')$ in \mathcal{A} . This equivalence follows straightforwardly from the definitions. \square

Theorem 4. *The generalized repeated reachability problem for POCAs is in PSPACE.*

Proof. We transform the given POCA into a 2WFA \mathcal{A} according to Lemma 1. It remains to check whether \mathcal{A} accepts some parameter word. For each variable $x \in X$, there is a 2WFA \mathcal{A}_x (even one-way) that checks whether x occurs exactly once. Then, $L(\mathcal{A})$ contains some parameter word iff $L(\mathcal{A}) \cap \bigcap_{x \in X} L(\mathcal{A}_x) \neq \emptyset$. Thus, according to Theorem 3, the problem is in PSPACE. \square

Remark 2. The original model of POCAs from [1] includes parameterized inequality tests of the form **less-than**(x) or **greater-than**(x), though they are not needed for the model checking result [2]. Using alternation, our method extends to that generalized model. For example, to simulate **less-than**(x), we spawn an automaton that checks

whether, in the input word, there is an occurrence of x on the right-hand side of the current position. Furthermore, it is easily seen that our approach can be used to decide the *reachability problem* in POCAs, too: Given two configurations $(q, i), (q', i') \in Q \times \mathbb{N}$ (where i and i' are encoded in unary), do we have $(q, i) \Longrightarrow_{\gamma}^* (q', i')$ for some γ ?

Since model checking flat freeze LTL on one-counter automata can be reduced to generalized repeated reachability in a parameterized one-counter automaton of exponential size [1], it follows that model checking flat freeze LTL on unary one-counter automata is in EXPSPACE. We refer to [1, 2] for the definition of that problem.

It is unlikely that the EXPSPACE upper bound is tight. Future work may consist in a careful adaptation of the algorithm from [3] to the specific setting of POCAs.

REFERENCES

- [1] S. Demri and A. Sangnier. When model-checking freeze LTL over counter machines becomes decidable. In *Proceedings of FoSSaCS'10*, volume 6014 of *Lecture Notes in Computer Science*, pages 176–190. Springer, 2010.
- [2] A. Lechner, R. Mayr, J. Ouaknine, A. Pouly, and J. Worrell. Model checking flat freeze LTL on one-counter automata. In *Proceedings of CONCUR'16*, volume 59 of *LIPIcs*, pages 29:1–29:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.
- [3] O. Serre. Parity games played on transition graphs of one-counter processes. In *Proceedings of FoSSaCS'06*, volume 3921 of *Lecture Notes in Computer Science*, pages 337–351. Springer, 2006.